# An Interpretable Approach to Fake News Detection

Caio Brighenti
Department of Computer Science, Colgate University

April 23, 2020

**Abstract**

The text of your abstract. 200 or fewer words.

*Keywords:*

# 1 Introduction

Propaganda has long been a tool of political influence, but in recent years it has taken a new online form: fake news. Fake news, once a buzzword on the internet, is now at the center of global politics, one of the most common bigram in the lexicon of United States president Donald Trump. After the term gained prominence in Trump's presidential campaign in 2016, it exploded into public conciousness, earning the distinction of Webster-Collins' "Word of the Year" in 2017.[1] As the current presidential race unfolds, fake news has returned to the center of the conversation, with major social media companies facing scrutiny of their misinformation policy. This phenomenon is also not a distinctly American problem–investigate reporting both during and after the 2018 Brazilian president election demonstrated that more than 40% of right-wing viral news articles shared on the popular messaging service WhatsApp were fake news favoring the eventual winner, Jair Bolsonaro.[2]

Fake news is not only politically significant but also dangerously tempting. Studies have shown false content propagates faster through social media than real content.[3] Blatantly false or exaggerated rhetoric can even lead to violent action, as demonstrated by the "Pizzagate" incident in which a man stormed a D.C. pizzeria with an AR-15, having been convinced by false and unverified information that a pedophile ring operated out of the restaurant's basement.[4] Fake news can also be incredibly easy to create. In 2019, a group of researchers at the Allen Institute for Artificial Intelligence published text generation model able to produce fake news.[5] In a troubling conclusion, the researchers found that state-of-the-art fake news detection systems struggled more with identifying fake news produced by their systems than actual fake news.[6] Fake news is thus easy to create, spreads quickly, and is hard to detect, a dangerous combination making it a serious threat to civic society.

Given the danger that fake news poses, machine learning and natural language processing researchers have devoted significant attention to the problem of fake news classi-

[1] Flood (n.d.)

[2] Avelar (n.d.)

[3]

[4]

[5]

[6]

fication. However, previous attempts at fake news classification overwhelmingly rely on highly complex models suffering from the "black box" problem. As a result, these models lack interpretability and do not allow us to reach new conclusions about the nature of fake news. In order to begin closing this gap, this research adopts an interpretable approach, with the overall objective of a producing a fake news classification model with comparable accuracy to state of the art models without compromising interpretability.

PARAGRAPH SUMMARIZING FINDINGS

# 2 Prior Work

Since the 2016 U.S. presidential election, fake news has been a frequent topic of natural language processing research. There are countless examples of papers approaching fake news classification or closely related problems, but from slightly different angles. This section summarizes the prior work in fake news detection, clarifying the different categories of methods. In general, previous works in fake news detection differ in three major ways: 1) the scale of the predicted variable, the information used as features, and the type of model. With respect to scale, any fake news detection model falls into one of three levels of granularity: 1) claim level, 2) source level, and 3) article level. These levels of analysis describe the response variable being predicted.

Claim level approaches attempt to determine whether a given claim, usually one or several sentences, is true or intentionally misleading. [7] Given that claim-level approaches must make a judgement based on only a short amount of text, researchers often adopt a fact-checking strategy This strategy, also known as "truth discovery," assumes that a sentence's claims can be gramatically isolated and checked against a database of established claims.[8] A natural application for claim-level models is social media, most commonly Twitter, where little is known about the author and only a very limited amount of text is available. However, claim-level approaches have serious limitations, often struggling with the complex sentences journalists or other writers typically employ.[9] Aditionally, they rely

[7] Examples of claim-level approaches include ...

[8] strube p. 2

[9] strube 2

entirely on a complete knowledge base, which must be constantly expanded and updated, clearly a difficult task.

Source level approaches attempt to classify whether a speaker or entire news source consitently publish misinformation. The intuition behind these approaches is that speakers or sources that have published misinformation in the past are likely to continue to do so. An example of a source-level approach is the popular browser extension "BS Detector,"[10] which classifies articles on a fine-grained scale of veracity by checking the source's status in a database of news sources and their reliability. A source's history of misinformation can also be used as a predictor in claim-level or article level approaches. Kirilin and Strube, for instance, create *Speaker2Credit*, a metric of speaker credibility, and show how it can improve the performance of fake news detection models when used as an input.[11]

Article level approaches have received the most attention in the work on fake news detection. This is logical, given that fake news tends to take the form of articles, peddling misinformation in the article text while posing as a legitimate source. Article-level approaches also benefit from a rich list of predictors to choose from, including not only the article's content but all relevant metadata. Kai Shu et. al, for instance, build an article-level model using linguistic and visual components of the article content, the social context around it—including information on the user that posted it, the post itself, responses to it, and the social network of the poster—, as well as spatiotemporal information capturing when and where the article and responses were to it were posted from.[12]

The work of Shu et. al is an example of the overwhelming number predictors available to researchers working in fake news detection, resulting in a diversity of approaches with respect to feature selection. Melanie Tosik et. al, for instance, employ only hand-crafted features capturing the similarity between an article's title and text in a two-stage ensemble classifier modeling whether an article's body agrees with its headline.[13] Sonam Tripathi and Tripti Sharma demonstrate the effectiveness of parts of speech tagging—also known as grammatical tagging—in document classification problems, the general category of nat-

[10]

[11] Kirilin and Strube

[12] Shu et al

[13] Tosik et al

ural language processing that article-level fake news detection falls under.[14] Ramy Baly et al. employ a breadth of features to model factuality and bias of news sources, using features covering the content of articles, the source's Wikipedia and Twitter pages, the structure of the URL, and the source's web traffic. [15]

Independent of level of analysis or choice of predictors, researchers overwhelmingly choose to use complex deep neural networks. Ajao et. al, for instance, use a "hybrid of convolutional neural networks and long-short term recurrent neural network models" to classify Tweets as true or false based on their text content.[16] The dominance of deep learning approaches is visible in an extensive survey on fake news detection done by Ray Oshikawa and Jing Qian. [17] The pair's section on machine learning models dedicates a total of three sentences to "Non-Neural Network Models," compared to seven paragraphs. focusing on neural networks.

While deep learning approaches can produce highly accurate models that consistently succeed in identifying misinformation, they also suffer from a lack of interpretability. This is often referred to as the "black-box" problem, meaning that the inputs and outputs of these models are perfectly clear, but the steps that the model takes to reach the output are completely invisible. This has been identified as a limitation of the the work on fake news detection thus far.[18] Oshikawa and Qian, at the conclusion of their extensive survey, declare that "we need more logical explanation for fake news characteristics," highlighting the need for models that can teach us something about fake news.

Approaches that focus on interpretability are rare, but do exist. From the deep learning perspective, Nicole O'Brien et al. employ post-hoc variable importance to their text-based deep learning model, identifying the words that are most predictive of fake and real news.[19] Their approach, however, does not interpret the results, but instead merely demonstrates the feasiblity of the technique. Furthermore, this method reveals only information about *specific* words, as opposed to *types* of words. More applicable is the work of researchers

[14] tripathi

[15]

[16] ajao et al

[17] Oshikawa

[18] shu, O'Brien

[19] O'Brien et al

who both use features that describe the semantic properties of the text in general, use interpretable models, and extensively document their results. The best examples of this type of work are the works of Benajmin Horne et al. and Mauricio Gruppi et al., both of which employ features capturing the complexity, style, and psychology of fake news, and display precisely how fake and real news differs in each of the variables used.[20]

# 3 Methodology

This paper seeks to contribute to the small literature of interpretable fake news detection, by following the methodology of Horne et al. and Gruppi et al., leveraging features that describe textual properties of fake news, applying non-neural network models and focusing heavily on interpretation of results. This section details the precise methodology employed, discussing the dataset used, the feature engineering process, outlier removal, and models applied.

There are many datasets freely available for fake news detection, but many suffer quality limitations. Oshikawa and Qiang outline 12 requirements for a quality fake news dataset, expanding a 9-point list originally by Rubin et al.: 1. Availability of both truthful and deceptive instances; 2. Digital textual format acessibility; 3. Verifiability of ground truth; 4. Homogeneity in lengths; 5. Homogeneity of writing mattes; 6. Predefined timeframe; 7. The manner of news delivery; 8. Pragmatic concerns; 9. Language and culture; 10. Easy to create from raw data; 11. Fine-grained truthfulness; 12. Various sources or publishers.[21] While no dataset currently exists that meets all 12 criteria, the article-level *FakeNewsNet* (FNN) dataset meets most.[22]

FNN is an article-level dataset that includes the title and body of each article (2), each of which have been profesionally fact-checked and labeled as true or false by Politifact or Gossicop(1, 3). FNN provides both political and celebrity news articles, but this paper chooses to use only the political articles in order to maintain a roughly consistent corpus (4,5,7). These articles are all in English, largely center around American politics, and

[20]Horne, Gruppi

[21]Oshikawa, Rubin

[22]FNN

come from a variety of sources (9, 12). Finally, there is little work needed to obtain the dataset, as FNN provides an API to quickly obtain the body and title of each article (10). The biggest limitation for FNN is that it lacks a fine-grained scale of truth, labeling only as binary true/false (11). However, given that it meets the majority of the criteria, and contains 726 observations, it is overall a good fit for this paper.

While FNN includes a host of metadata on each article, this paper utilizes only features engineered from the text and titles of each article. The objective of this paper is to reach new conclusions about the content of fake news, making certain metadata irrelevant or problematic. For instance, the usage of website traffic to gauge veracity may increase accuracy,[23] but measures nothing about the actual content of the article. It should be obvious that websites with high traffic are perfectly capable of producing misinformation. Aditionally, equating established sources with reliable sources can be problematic, failing to hold mainstream media accountable and preventing the growth of new, quality publishers.

The choice of using features capturing exclusively textual properties is motivated by a belief that the problem of fake news cannot be solved with deep learning models automatically policing all the content on the internet. A widespread system like this would have incredible power, and could easily become a force for opression and misinformation with biased data or improper use. Aditionally, people are unlikely to be convinced that something they believe to be true is misinformation just because a browser extension tells them it is. Fake news is a social problem, and will only be fixed with widespread education on how to identify fake news. To achieve this, this paper treats fake news detection as a learning opportunity, focusing solely on identifying the textual properties of text that might suggest an article is misinformation.

This paper uses the FNN dataset to obtain the title and body of articles as well as a true/false label, then leverages a series of natural language processing tools to engineer features describing the text. Each feature is calculated for both the body and the title seperately. Each feature falls under one of four categories: 1) complexity metrics, 2) summary, grammatical, and psychological metrics from the Linguistic Inquiry and Word Count engine [24], 3) parts-of-speech tagging, and 4) named entity recognition. Each cate-

[23] forget which one does this

[24]

gory represents a different method or tool for feature engineering. Each feature engineered is displayed in the tables below.[25].

The complexity metrics are calculated in three different ways. Several of these metrics are indexes of textual complexity calculated using the `quanteda` package in R. Others are variables describing the structure of verb-phrase and noun-phrase trees obtained for each sentence using the Stanford CoreNLP constituency parser. The final complexity metric is a manually computed type-token ratio, where types are all the unique words in a document and tokens are the total words in that document, capturing the diversity of the vocabulary used.

The LIWC features comprise a diverse range of different metrics, many capturing the different psychological components within the text such as cognitive processeses, or core drives and needs.[26]. Some LIWC variables capture simpler properties, such as the frequency of informal speech or specific punctuation marks. As LIWC's metrics are heavily dependent on counting words from dictionaries, each metric is normalized by the text of the document, providing a sense of how frequently types of words occur in a standard document size.

All features falling under the third and fourth categories were obtained using the Stanford CoreNLP toolkit. Specifically, the grammatical incidence variables were obtained using the CoreNLP parts-of-speech tagger, which counts the frequency of types of words (for instance, verbs) within a document. As before, these metrics were normalized to account for differing document lengths. Finally, the named entity recognition feature is obtained using the CoreNLP named entity recognition annotator, which simply counts the total number of words that refer to named proper nouns.

The complete data thus consisted of 726 observations with 152 features each, constructed using the body and title of each articleo btained by the FNN API, which queries the stored article URLs. However, given that web pages often change structure, move to different addresses, or are removed from the internet entirely, many of the entries have changed since the dataset was initially compiled, and are now unavaiable or in an incorrect

[25]And described in detail in Appendix A

[26]For a full description of each of these variables, consult Appendix A or the LIWC Operator's Manual available at

format and must be removed from the dataset. To identify outliers, a baseline logistic regression model using all features was fit in order to identify overly influential observations with Cook's Distance four times larger than the mean. This approach cannot perfectly identify all outliers, but suggests that these observations are *potential* outliers. Each of the 575 potential outliers were manually inspected and labeled as either false positives or true outliers. Out of the 575 potential outliers, 422 were true outliers, 132 of which were Politifact articles, bringing the number of observations down to 594.

After having completed all feature engineering and outlier removal, exploratory data analysis was performed to identify group differences in each predictor between true and false articles. This approach reflects the work Horne et al. and Gruppi et al., allowing for the comparison of results between shared predictors. However, instead of applying an ANOVA test to each predictor, a Mood's Median hypothesis test was performed using the `RVAdeMemoire` package[27] as many of the predictors did not meet the normality assumption required by ANOVA and other traditional tests. Aditionally, estimates and 95% confidence intervals for each predictor accross both groups were calculated using bootstrapping 1,000 samples.[28]

Given the extensive number of predictors and the multicolinearity between many of them, a binomial Lasso regression—also known as logistic regression with L1 regularization—was applied to avoid overfitting by creating a more parsimonious model. Aditionally, to avoid skewed accuracy results due to class imbalance, the dataset was upsampled to have an equal proportion of negative and positive labels. The upsampling increased the number of negative articles to have a 374-374 split, as opposed to the original 374-220 split. After fitting the Lasso model, the features reduced to zero were removed and a logistic regression model using only the preserved features. This is the model used for final interpretation of results. Aditionally, seperate models are fit for the body and title of articles. An approached focused entirely on predictive accuracy would likely instead create a two-stage ensemble model, but as the overall objective is interpretation, maintaining the two seperate is preferable as it allows for better interpretation of the results.

[27]

[28]The results of the Mood's Median tests are displayed in the next section, but the confidence intervals are only shown in Appendix A.

# 4 Results

This section summarizes the results of the analysis, starting with the pairwise Mood's Median tests followed by the results of the modeling process. Results are compared with the work of Horne et al. and Gruppi et al. to highlight overlaps and disagreements between their results and the results of this paper.

## 4.1 Mood's Median Tests

# 5 Discussion

# 6 Future Work

# References

Avelar, D. (n.d.), 'Whatsapp fake news during brazil election 'favoured bolsonaro", *The Guardian* .

**URL:** *https://www.theguardian.com/world/2019/oct/30/whatsapp-fake-news-brazil-election-favoured-jair-bolsonaro-analysis-suggests*

Flood, A. (n.d.), 'Fake news is 'very real' word of the year for 2017', *The Guardian* .

**URL:** *https://www.theguardian.com/books/2017/nov/02/fake-news-is-very-real-word-of-the-year-for-2017*